

Objectives:

1. Implementing a system for "Data Collection and Processing"

Description:

Write a python/C/C++/Java code in a modular* approach that takes input from the console. The console input may contain a set of switches as parameters followed by the program name and their corresponding parameter values. Let's name the program "DCP" which stands for "Data Collection and Processing ". Your program should be able to handle at least the following switches:

Parameters/Switches:

1. -a algoname (the algorithm name can be either FP or AP where FP is a function that should flip patterns of texts by reversing them and AP is a function that should append patterns of the same type. See below for more detail about FP and AP)
2. -d datasetName (datasetName is the name of the dataset. Default dataset is "toy.txt" which is a text file containing the dataset given in Han's book. The datasets that are to be tested are Mushroom, Kosarak, Chess, and Retail found in <http://fimi.uantwerpen.be/data/>)
3. -p partition (a partition is a floating-point number stating the size of the data to be used for processing. By default this is 0.5 which means 50%.)
4. -m (this switch requests your program to store memory Consumption, i.e., the amount of memory used in MB by a particular run of an algorithm with the given parameter setting, i.e. partition size, dataset and processing algoname.)
5. -rt (this switch request your program to store runTime, i.e., the amount of time required in a millisecond by a particular run of an algorithm with the given parameter setting, i.e. partition, dataset and algoname.)
6. -o outputFileName (outputFileName represents the CSV file name where the threshold will be written in the 1st column, and the time required and memory usage will be written in the next two columns. Note, if this switch is not specified explicitly then by default the same file will be used to append for the same algorithm and same dataset. Otherwise, a new file has to be created where each file should have the following name pattern the values for switch "-p" plus an underscore "_" then the value for the switch "-d" and ".csv".)

NB:

1. In order to save your time from waiting for outputs, you can write a batch file for windows or a shell script for Linux/mac where all the runs with required parameters or switches are listed. Once you run the script or batch from a command line, you may set out for other works leaving your computer dealing with the computation by itself. Just make sure no one turns your computer off while the batch/script is running. For your reference, a sample batch/script file is shown below.
2. Your program must include all kinds of exception handling.
3. In case of an invalid switch or invalid value for a particular switch, your program should show a helpful hint instead of treating the value in an inappropriate way or terminating abruptly.
4. The FP will reverse every word and numeral. E.g., if the input file contains "Hello 123.456" it will be "olleH 654.321"

5. The AP will append similar data types together. E.g. "Hello world, I am 12 34 45 Not very good" will be "Helloworld,Iam 123445 Notverygood".

Sample Batch file content with potential inputs:

Python DCP.py -a FP -d mushroom.txt -p 0.3 -rt -m

Python DCP.py -a AP -d mushroom.txt -p 0.3 -rt -m

.sh => for linux and mac

.bat => for windows